

ASSIGNMENT INFORMATION

Points Possible

5

We've looked at how to attack substitution ciphers. An affine cipher is a substitution cipher, but it seems a bit more complex. Can you think of another substitution cipher that might be harder to break than, say, the Caesar cipher? Why do you think it's more difficult to break? Implement the cipher you've designed in Python and submit the cipher to me as a single python file (e.g. a script, using good program conventions). I suggest you implement it as a single function, as we did in the examples. Have the function take a plaintext string and return a ciphertext string (and vice versa). Any string consisting of letters and spaces should be acceptable. If I can't submit a string like that, I'll bounce the assignment back with a zero for another submission. Don't forget to convert the string to upper case for your cipher. Returning an all uppercase ciphertext is just fine. **Remember, this must both encrypt and decrypt.**

Note: You don't need to implement interactivity with this program. A program like this is fine, you implement the encipher() and decipher() functions, and any other functions you'd like:

```
def encipher(plaintext):
    # Your code

def decipher(ciphertext):
    # Your code

def main():
    plaintext = 'MACARONI'
    ciphertext = encipher(plaintext)
    print("{} {} : {}".format(plaintext, ciphertext, decipher(ciphertext)))

if __name__ == '__main__':
    main()
```

Evaluation criteria: I'm looking for these things in this assignment:

100%: The cipher works, and it's one we haven't covered.

50%: The python program runs, but the cipher doesn't seem to work.

0%: You don't submit a cipher.

Good luck! I look forward to your submission!